REMARKS

Claims 2-4, 7-19, 21-33, 35-44, 46, and 47 are pending. Claims 1, 5, 20, 34, and 45 are canceled without prejudice. This Amendment is in response to the Office Action mailed August 2, 2007 (hereinafter, "Office Action").

Amendments

The specification is amended to correct minor informalities noted upon review of the specification.

The claims are generally amended for clarification. Claim 6 is amended to generally place in independent form. Claims 35-44, 46, and 47 are amended to set forth Beauregard style claims rather than system claims.

No new matter is introduced by this Amendment.

Claim Rejections - 35 U.S.C. § 101

Claim 1 stands rejected under 35 U.S.C. § 101. Applicants respectfully submit that the Office Action improperly supports this rejection by stating that the claims were "unclear," which is a ground for rejection under 35 U.S.C. §112, second paragraph, not 35 U.S.C. § 101. Nevertheless, by this Amendment, claim 1 is canceled thereby obviating this rejection. Withdrawal of this rejection is therefore respectfully requested.

Claim Rejections - 35 U.S.C. § 103(a)

Claim 2 stands rejected for being obvious under 35 U.S.C. § 103(a) in view of U.S. Patent 6,021,510 issued to Nachenberg (hereinafter "Nachenberg '510") in view of U.S. Patent 5,826,013 also issued to Nachenberg (hereinafter "Nachenberg '013"). In addition, although not explicitly rejected, claims 3-47 were addressed under the heading "Claim Rejections – 35 USC 103." For the purpose of this Amendment, it will therefore be assumed that all claims 2-47 were intended to be rejected under 35 U.S.C. § 103(a).

Applicants respectfully traverse rejections of claims 2-47 under 35 U.S.C. § 103(a) because the prior art fails to teach or suggest each and every feature set forth in the claims, and because the Office Action failed to clearly explain how a person of ordinary skill in the art would have been motivated to make the changes and/or modifications of the prior art to meet each and every feature of the claims.

For obviousness under 35 U.S.C. § 103(a), each and every limitation must be taught or suggested by the prior art reference, or references when combined or modified (MPEP 2143). It should therefore be noted that Applicant need only point out a single limitation in each claim that is not disclosed, taught, or suggested by any reference identified in the Office Action to overcome the prior art-based rejections. The following discussion therefore should not be construed as an exhaustive listing of every distinguishing feature set forth in the claims.

Claims 2 and 35 are directed to a method and a tangible medium embodying program instructions causing a computer to carry out a method respectively, where the method includes, *inter alia*, "identifying a next instruction to be executed when executing a series of instructions" and "for the next instruction, determining an identifying value for a current memory block that contains the next instruction."

The Office Action relies on Nachenberg '510 to show "determining an identifying value . . ." (Office Action, page 3, lines 8-9[1]), but admits that Nachenberg '510 "does not explicitly state that the current instruction is verified dynamically before being executed." Applicants agree that the Nachenberg '510 doesn't mention or suggest dynamic verification. However, for the dynamic verification feature, the Office Action turns to Nachenberg '013, which, according to the Office Action, "discloses examining the instruction/interrupt usage profiles of each known polymorphic virus as each instruction is fetched for emulation" (last two lines of page 3 of the Office Action).

While Applicants agree that Nachenberg '013 suggests comparing fetched instructions with an instruction/interrupt usage profile, Applicants respectfully disagree that (1) this meets the claim limitations and (2) to the extent that it does meet the claim limitations by combining and/or

[1] All printed lines in source documents, except documents already containing line numberings (e.g., issued patents), are counted, including headings, but not including the page header. White space is not counted.

modifying the references as necessary under 35 U.S.C. § 103(a) to meet the claim limitations, the prior art lacks motivation to make the combination and/or modification.

Nachenberg '510 is directed to anti-virus file scanning. As generally known in the art, it is common to scan an entire file accessed by an application or operating system to ensure it contains no malicious code prior to executing the file. Nachenberg '510 describes a method for maintaining a list of data blocks on a disk that are known to be free of viruses, so that files formed by the disk blocks need to be scanned only once each time the block is written or the virus signature database is updated. Nachenberg '510 makes no provision for performing the antivirus scanning "when" executing the files. In fact, the Office Action acknowledges that Nachenberg '510 fails to teach or suggest "that the current instruction is verified dynamically before being executed" (Office Action, page 3, lines 20-21). As mentioned above, however, the Office Action relies on Nachenberg '013 for "examining the instruction/interrupt usage profiles of each known polymorphic virus as each instruction is fetched for emulation" (Office Action, page 3, lines 21-22).

Nachenberg '013 is directed to an improved mechanism to identify polymorphic viruses. Polymorphic, or "mutating" viruses thwart static virus scanners that simply compare a virus signature with a static file to identify the virus by encrypting a portion of the virus with a random encryption key (col. 1, lines 18-40). As shown and described by Nachenberg '013 with reference to Figure 1C, polymorphous viruses have a decryption routine 164 that decrypts and passes control to a static virus body 160. Nachenberg '013 teaches emulating execution of instructions within a file for the purpose of running the virus' own decryption routine to expose a static virus body of a polymorphic virus prior to scanning the file (col. 6, lines 3-17). In the improvement described by Nachenberg '013, as each instruction is fetched for emulation of the decryption routine, the fetched instruction is compared with an instruction/interrupt usage profile for each known polymorphous virus (col. 8, lines 24-27). An instruction/interrupt usage profile identifies specific ones of the 256 basic types of instructions available in x86 architecture that are used by a certain virus (col. 8, lines 45-65). Thus, as each instruction is fetched, non-matching viruses can be flagged to remove them from consideration as a potential match. If all the polymorphous viruses are flagged, than either the file is not a known polymorphous virus, or the decryption portion of the polymorphous virus is completed (col. 9, lines 33-38). Then, once emulation is

terminated, scanning can begin on the (at least partially) decrypted static virus body (col. 9, lines 50-52).

Each of claims 2 and 35 have been amended. Thus, claim 2 now provides "identifying a next instruction to be executed *when executing a series of instructions*" (emphasis added) and "for the next instruction, determining an identifying value for a memory block that contains the next instruction." (Support for these changes are found, for example, in the specification at paragraph 30.) Thus, the operations of "identifying a next instruction" and "determining an identifying value for a memory block that contains the next instruction" are linked. One cannot determine an identifying value for a memory block that contains the next instruction without first identifying the next instruction, and in the case of claims 2 and 35, the next instruction is identified "when executing a series of instructions," which is what makes the verification *dynamic*.

Nachenberg '510 teaches a mechanism of tracking disk sectors for files that have already been scanned so that the same files need not be scanned unless the file (i.e., disk blocks making up the file) is modified or the virus definitions are updated. Nachenberg '510 mentions determining an identifying value, but not dynamically, and not for "a next instruction" to be executed. Nachenberg '013 teaches a method of scanning files to identify polymorphous viruses which involves fetching a next instruction and dynamically doing *something*, but that *something* is emphatically not "determining an identifying value for a memory block that contains the next instruction." Thus, neither of the references, suggest, "determining an identifying value for a memory block that contains the next instruction."

In making the modification/combination proposed by the Office Action in accordance with obviousness analysis under 35 U.S.C. § 103(a), it appears that the only concept contributed by Nachenberg '013 in the combination contemplated in the Office Action is dynamically doing something when an instruction is fetched/identified. The Office Action simply took the concept of dynamically doing something, and applied it to Nachenberg '510, asserting that it would one would have been motivated to dynamically verify a disk block in which a next instruction resides. However, the prior art does not provide such motivation.

More particularly, the Office Action states that "it would have been obvious . . . to modify the method of '510 with the examining of instructions after being fetched in order to flag polymorphic viruses that deploy mutation engines" (Office Action, page 4, lines 1-3). Applicants agree that one of ordinary skill in the art may have been motivated to identify polymorphous viruses as taught by Nachenberg '013 when implementing the scanning engine of Nachenberg '510. However, even when such a combination is made, e.g., by inserting the procedure of for identifying polymorphous viruses as described in Nachenberg '013 into antivirus scan module 3 shown and described with reference to Figure 1 in Nachenberg '510, the operation of "determining an identifying value for a memory block that contains the next instruction" would still not be performed.

Furthermore, the Office Actions fails to adequately explain how a person would be motivated to combine and/or modify the references as necessary to dispense with the key operation of Nachenberg '013 of "compar[ing] the fetched instructions with an instruction/interrupt usage profile" and replace it with "determining an identifying value for a memory block that contains the next instruction." For this reason, Thus, Applicants respectfully submit that the Office Action fails to satisfy the explicit analysis required to sustain a rejection as set forth in *In re Kahn*, 441 F. 3d 977, 998 (Fed. Cir. 2006) as recently cited with approval by the United States Supreme Court in *KSR International, Co., v. Teleflex, Inc.*, 127 S.Ct. 1727, 1741 (2007) (slip op., page 14). Thus, the Office Action fails to make out a *prima facie* case for obviousness.

For the above reasons, Applicants respectfully submit that claims 2 and 35 are patentable over the Nachenberg references because claims 2 and 35 provide that the next instruction is determined "when executing a series of instructions" and "for the next instruction, determining an identifying value for a memory block that contains the next instruction" ... "whereby the next instruction is verified dynamically before being executed." Independent claims 2 and 35 should therefore be allowed.

Dependent claims 2-4, 8-19, 21-33, 36-44, 46, and 47 depend from either claim 2 or claim 35 and therefore should be allowed for at least the same reasons as claims 2 and 35. Furthermore, dependent claims 2-4, 8-19, 21-33, 36-44, 46, and 47 further define and distinguish the invention from the prior art for reasons that do not need to be addressed herein.

Claim 6 is written into independent form (and is otherwise amended for clarification purposes) and includes a feature of "wherein computing . . . the hash value comprises applying a mask to the current memory block, the mask being a data structure that designates at least one byte of the current memory block be ignored in the computing of the hash value, the data structure designating less than an entire memory block so that the hash value is based on only part of the contents of the current memory blocks." The Office Action identifies Nachenberg '510 as essentially teaching this feature at column 5, lines 6-33 (Office Action, page 5, lines 12-13). This portion of Nachenberg '510 discusses "scanning and hashing a minimal set of sectors in [the] file." As explained in more detail at col. 2, lines 43-62, for some file formats, only certain areas of a file may indicate the presence of a virus. In col. 3, lines 61-65, it is explained that less than all the sectors of a file may be scanned an initial time. And in col. 4, lines 40-47, upon rescanning, hash values for previously scanned sectors are compared with newly-computed hash values for those sectors to determine if the important areas of the file have been modified since the previous scan. This procedure outlined in Nachenberg '510 improves performance by "hashing a minimal set of sectors" (col. 5, lines 6-7).

In contrast, claim 6 is directed to "applying a mask to the current memory block, the mask being a data structure that designates at least one byte of the current memory block to be ignored in the computing of the hash value . . ." (claim 6, lines 13-14). Thus, claim 6 includes a feature of "designating less than an entire memory block so that the hash value is based on only part of the contents of the current memory block" (last three lines of claim 6). This distinguishes the invention as defined by claim 6 from Nachenberg '510 because in Nachenberg '510, each sector (which correspond to "memory block" when referring to data stored on disk) is either hashed or not, and when hashed, the entire contents of the sector is used to calculate the hash value.

Applicants therefore respectfully submit that claim 6 contains features that are neither taught nor suggested by the prior art of record, and therefore, that claim 6 should be allowed. Since claim 7 depends from claim 6, claim 7 should be allowed for at least the same reasons as claim 6.

For the reasons expressed above and others, Applicants respectfully submit that the present Application is in condition for allowance. Applicants therefore respectfully request reconsideration of the outstanding rejections and a Notice of Allowance. The Examiner is invited to contact the undersigned at 650-427-2390 to discuss any additional issues that may be resolved by Examiner's Amendment or otherwise in order to bring this prosecution to a rapid close.

Date: February 4, 2008

Respectfully submitted,

Leonard Heyman
Reg. No. 40,418
Attorney for the Applicant